

## AMENDMENTS TO THE SPECIFICATION:

Please replace the **paragraph [006]** spanning pages 2-3 in the specification with the following replacement paragraph:

-- The Jini™ architecture has been developed using the Java programming language to enable devices in a distributed system to share services using remote method invocation (RMI). Traditional Jini systems use RMI to enable a client device to request and receive a service provided by a server device on a remote machine. While conventional Jini systems provide a basic architecture for providing services in a distributed system, they do not provide tools specifically directed to providing complex services. Current systems do not address provisioning a service, such as application software, to make it available to the distributed system in the first place. Furthermore, conventional systems do not consider the ~~requires~~ requirements of a specific service before provisioning the service to make it available in the distributed system.

Please replace the **paragraph [074]** spanning pages 25-26 in the specification with the following replacement paragraph:

-- A queue 1508 (not shown) may be implemented as both an event listener and an event producer. In this way, for example, queue 1508 may receive events from adapter 1502 and send events to filter 1504 or logger 1506. Queue 1508 may include a unique name, a Java™ class that implements the filter, a set of properties, and a set of outputs. The properties may be, for example, a sequence of name value pairs that are passed to the queue's constructor. The outputs may be, for example, a sequence of component names that will be registered as event listeners and/or event producers to

the queue. The outputs may designate, for example, adapters 1502, filters 1504, or loggers 1506. The following is program code for an exemplary configuration of ~~filter~~ queue 1508 written as an XML document:

```
<ems:queue>
  <ems:name>EventQueue </ems:name>

  <ems:classname>com.sun.autoid.queue.MultiThreadedQueue</ems:classname
>
  <ems:properties>
    <ems:property>LogLevel</ems:property>
    <ems:value>INFO</ems:value>
  </ems:properties>
  <ems:properties>
    <ems:property>MaxThreads</ems:property>
    <ems:value>10</ems:value>
  </ems:properties>
  <ems:outputs>
    <ems:output>EventLogger </ems:output>
    <ems:output>FileLogger</ems:/output>
  </ems:outputs>
</ems:queue>
```

Please replace the **paragraph [075]** spanning pages 26-27 in the specification with the following replacement paragraph:

-- In an embodiment of the present invention, logger 1506 may be implemented to notify user 1306 of data from readers 1302. Logger 1506 may notify user 1306 of RFID and non-RFID events using a protocol specified by user 1306, for example, by logging information to a file system, a JMS queue, or a XML/HTTP file. Logger 1506 may function as an event listener (e.g., for data from filters) and/or an event producer (e.g., for data for users). Logger 1506 may include, for example, a unique name, a Java™ class that implements the logger, and a set of properties. The properties may be, for example, a sequence of name value pairs that are passed to the logger's

constructor. The following is program code for an exemplary configuration of filter

logger 1506 written as an XML document:

```
<ems:logger>
  <ems:name>EventLogger</ems:name>
  <ems:classname>com.sun.autoid.logger.REProduce</em
s:classname>
  <ems:properties>
    <ems:property>LogLevel</ems:property>
    <ems:value>ALL</ems:value>
  </ems:properties>
  <ems:properties>
    <ems:property>EventID</ems:property>
    <ems:value>97531</ems:value>
  </ems:properties>
</ems:logger>
```